

## **IBM PROJECT**

### **Create Node-RED Service**

TEAM ID	<b>PNT2022TMID13304</b>
TEAM MEMBERS	S. Manikandan, P. Manikandan, M. Deepak, M. Richard Kumar. K. Dharnesh
PROJECT NAME	IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE.

### **Learning objectives:**

In this tutorial, you will learn how to:

- Create a Node-RED starter application running in the IBM Cloud
- Secure the application
- Customize the Node-RED Starter Kit by adding additional nodes

### **Prerequisites:**

To complete this tutorial, you need an IBM Cloud account.

This tutorial requires an IBM Cloud Pay-As-You-Go account. To upgrade your Lite account, go to your account settings. In the Account Upgrade section, click **Add credit card** to upgrade to a Pay-As-You-Go account, or click **Upgrade** for a Subscription account. See [Upgrading your account](#) for more information.

This Node-RED starter tutorial provides instructions on deploying the app to IBM Cloud Code Engine, which is a fully managed, serverless platform that runs your containerized workloads and manages the underlying infrastructure for you. IBM Cloud Code Engine provides

100000 vCPU seconds per month at no charge. Your Node-RED flow will often scale to 0, which means that you won't incur any charges for light to moderate usage. Review your consumption and confirm your billing on a regular basis.

## **Estimated time:**

## **Steps**

1. Find the Node-RED Starter Kit in the IBM Cloud catalog
2. Create your application
3. Enable the Continuous Delivery feature
4. Open the Node-RED application
5. Configure your Node-RED application
6. Add extra nodes to your Node-RED palette

### **Step1. Find the Node-RED Starter Kit in the IBM Cloud catalog**

1. Log in to IBM Cloud.
2. Open the catalog and search for **node-red**.
3. Click on the **Node-RED App** tile.

This will show you an overview of the Starter Kit and what it provides.

### **Step2. Create your application**

Now you need to create the Node-RED starter application.

1. On the *Create* tab, a randomly generated **App name** will be suggested. Either accept that default name or provide a unique name for your application. This will become part of the application URL.
2. The Node-RED starter application requires an instance of the **Cloudbant database service** with

IBM Cloud IAM and Cloudant credentials to store your application flow configuration. Select the region the service should be created in and what pricing plan it should use.

3. Click the **Create** button to continue. This will create your application and, if necessary a Cloudant database service instance, but it is not yet deployed to IBM Cloud.

### Step3. Enable the Continuous Delivery feature

1. On the next screen, click the **Deploy your app** button to enable the *Continuous Delivery* feature for your application.
2. On the next screen, click the **Code Engine** tile.
3. Scroll down after selecting the **Code Engine** tile. You will need to create an **IBM Cloud API** key to allow the deployment process to access your resources. Click the **New** button to create the key. A message dialog will appear. You can accept the default values and confirm to close the dialog.
4. Select the **Region** and **Container registry region**, to deploy your application to. This should match the region you created your Cloudant instance in.
5. Provide a unique **Project** name or accept the default 'project-name'

Click **Next** to continue.

6. Configure the **DevOps toolchain** by selecting the **region** it should be created in. Again, try to match the region you selected previously.
7. After a few moments, the **Deployment Automation** section will refresh with the details of your newly created Delivery Pipeline. The Status field of the pipeline will eventually show **In progress**. That means your application is being built and deployed.
8. The Deploy stage will take a few minutes to complete. You can click on the ci-pipeline **Status** link to check the progress of the Delivery Pipeline. Eventually the Deploy stage will display a green

checkmark and a **Success** message to show it has passed. This means your Node-RED starter application is now running.

#### Step 4. Open the Node-RED application

Now deployed your Node-RED application, let's open it up! May have to refresh your page.

On the application details page, you should now see the **App URL**, **Source** and **Deployment target** fields filled in.

Click on the **App URL** to open up your Node-RED application in a new browser tab.

#### Step 5. Configure your Node-RED application

The first time you open your Node-RED app, you'll need to configure it and set up security.

1. A new browser tab will open with the Node-RED start page.
2. On the initial screen, click **Next** to continue.
3. Secure your Node-RED editor by providing a **username** and **password**. If you need to change these at any point, you can either edit the values in the Cloudant database, or override them using *environment variables*. The documentation on [nodered.org](https://nodered.org) describes how to do this.  
Click **Next** to continue.
4. The final screen summarizes the options you've made and highlights the environment variables you can use to change the options in the future. Click **Finish** to proceed.
5. Node-RED will save your changes and then load the main application. From here you can click the **Go to your Node-RED flow editor** button to open the editor.

The Node-RED editor opens showing the default flow.

## Step 6. Add extra nodes to your Node-RED palette

The recommended approach is to edit your application's `package.json` file to include the additional node modules and then redeploy the application.

This step shows how to do that in order to add the **node-red-dashboard** module.

1. On your application's details page, click **Source** url. This will take you to a git repository where you can edit the application source code from your browser.

- 2.. Scroll down the list of files and click on **package.json**. This file lists the module dependencies of your application.

3. Click the **Edit** button

4. Add the following entry to the top of the dependencies section (1):

5. `"node-red-dashboard": "2.x",`

Add a **Commit message** (2) and click **Commit changes** (3)

6. At this point, the Continuous Delivery pipeline will automatically run to build and deploy that change into your application. If you view the Delivery Pipeline you can watch its progress. The Build section shows you the last commit made and the Deploy section shows the progress of redeploying the application.

7. Once the Deploy stage completes, your application will have restarted and now have the node-reddashboard nodes preinstalled.

## Summary:

Now created a Node-RED application that is hosted in the IBM Cloud. You have also learned how to edit the application source code and automatically deploy changes.